Modelling Ubiquitous Computing Applications

G.J. Doherty and M.D.Wilson
CLRC Rutherford Appleton Laboratory

Abstract

In this paper we look at some general approaches to modelling ubiquitous computing systems, and identify some key concepts and design issues dealt with by these approaches. Since the advent of these new systems raises a vast array of issues, we concentrate on models which help us to choose between design alternatives. We consider in turn models whose focus is on services provided, information manipulated, and the physical distribution of an application. As a standard design notation, we make reference to the Unified Modelling Language (UML) when considering construction of concrete models.

1. INTRODUCTION

One of the next major developments in computing will be the widespread embedding of computation, information storage, sensing, and communication capabilities within everyday objects and appliances (Weiser 1993, Norman 1998, Mittag 1999). Many prototype applications have been constructed and experimented with (Gershenfeld, 1999), and much of the required technological infrastructure is in place. These new systems, whether they be intelligent environments or "information appliances" will have a profound effect on the way we interact with computers and information, and ultimately the way we work and live. However, the design of these artefacts is potentially very different to traditional interactive system design. In this area, there are few constraints on the form of solution; there are very many different capabilities which may be embedded in an object; mobility and the ability to communicate and interact with other artefacts are very often requirements. Since in many cases the user interface is moved from a computer screen into the "real world", physical attributes of objects such as their location and proximity to other objects will be major factors in interactions with the user. The question we ask in this paper is how designers may take a considered and well−structured approach to the design of these artefacts and collections of artefacts, and particularly what modelling support is available. Hence, our focus is on models which allow us to compare design alternatives.

1.1 Ubiquitous Application Design

The challenges posed by this form of application are immense, but so too are the potential benefits of liberating the user from the constraints of contemporary desktop computing.

As with any application, the first challenge is to identify *what* it is to provide or accomplish, and how this will aid the user in carrying out his or her daily activities. Part of the difficulty here is the breadth of functionality which can be provided by a given device. (Norman, 1998) suggests that devices ("information appliances") will increasingly be focussed towards single tasks, although this is by no means universally accepted. Secondly, there is the question of *where* the application is to be accessible; users are generally mobile; by definition ubiquitous computing applications should support this mobility, either through omnipresence (as with wearable devices) or through a multiplicity of points of access (as in smart environments for example). The third question is *how* the application is to be instantiated; how functionality is to be physically distributed, and how existing infrastructure (physical devices, information services, communication networks and protocols, software toolkits and architectures) can be harnessed in providing this functionality.

In the following sections we look at some modelling approaches which address issues relevant to ubiquitous computing. We consider how such models might be encoded in a standard notation (UML), and identify issues where more support for modelling is needed. We would like to have available a set of models which aid in the evaluation of specific design alternatives. The aim of this is to move towards a reasoned approach to the design of ubiquitous computing applications.

2. MODELLING APPROACHES

We start by presenting three different approaches to modelling ubiquitous computing applications (with a focus on services, information and physical distribution, respectively) and the *design issues* which are addressed by each approach. Any practical approach to modelling must address to some extent the questions of what, where and how listed above, but their focus and level of abstraction can be very different.

2.2 Service Based Models

A very useful approach to applications in this domain is to view interactions between entities or actors in terms of services. This view holds between a user exploiting a service provided by an information appliance and also between an appliance and some lower level service or utility. An example of a service−oriented view can be found in (Nixon et al., 2000), which describes a smart environment implementation, where there are sensors for movement, location and state (for objects such as doors), and actuators controlling coffee machines, doors, telephones and so on. In addition to services provided by individual entities (eg. a printing service), there are three generic services. An event service provides a means for objects to communicate, a policy service which defines the capabilities of entities, and a management service responsible for carrying out the necessary tasks.

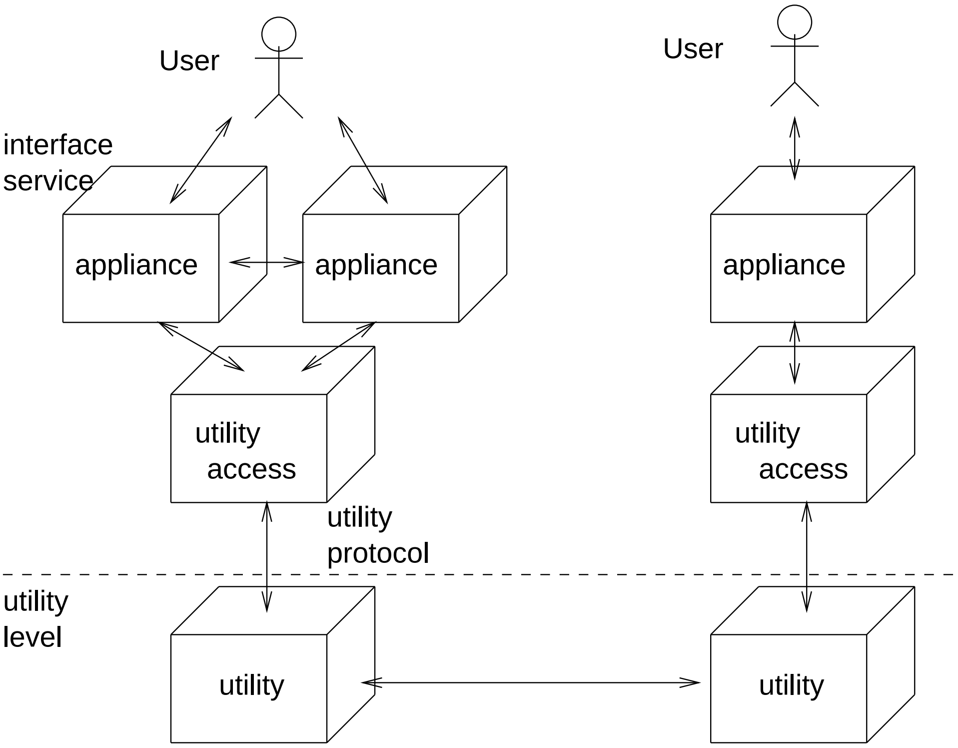

Figure 1 − Application and utility−level services

A service based concept shared by a number of researchers in this area is that of an *information utility*. This would be a common, ubiquitous resource, which is either omnipresent or with very many points of access, which can be tapped into when needed and with a uniform interface shared by many applications, much as electricity is provided in the modern world. We distinguish the utility level of service from an appliance−level service, firstly by its *ubiquity*, and secondly by it's *invisibility*. That is, the utility service is not directly visible or accessible to the user, who must access the utility through some appliance.

One could further distinguish appliance−level services by *intelligence*. We can look at intelligence along several dimensions:

− adaptation, context awareness; the device is "aware" of its location and objects in its surroundings. Context awareness (and subsequent adaptation) can be vital to maintaining continuity of service, and continuity with respect to a task. It can involve sensor fusion and activity detection.

− autonomy; whether user−meaningful decisions are made by the device without immediate consultation with the user.

Note however that this intelligence could be embodied either in appliance or in the utility, or in another entity accessed via the utility. For the user, it may not be necessary to know at what level the intelligence is implemented. A service model is useful as it accommodates both implementation possibilities. However, we must be careful of circumstances where the user could perceive a difference between services provided at different levels, for example with mobile applications where some utility exploited by the device might sometimes be unavailable or available at a degraded level.

From a modelling perspective, we need to have an information model if we are to address issues like scope of adaptation. Work on the modelling of agents has tended to focus on behavioural and protocol issues; representation of issues like responsibility deserves further attention.

An interesting aspect of this modelling approach is that we can view so−called "universal" appliances as providing an *interface service* to the user, exploiting the services provided by other appliances and utilities. Thus at the user level, the appliance is simply adding a physical interface to lower level services. In modelling terms, many of the issues here are concerned with interactive systems in general; how to distinguish and reason about user perceivable data and user−generated input. While we do not go in to this here, there are also new issues concerning continuous information exchange between system and user (Doherty and Massink 1999), use of novel input devices (Doherty et al. 2001) and so on.

Modelling Services and Quality of Service

The nature of the service itself, for example whether it is providing access to information or communication with other systems and users, is a difficult aspect to express within any precise framework, simply because of the very wide variety of possible services, and the various levels of abstraction at which these may be expressed.

For modelling services, the UML (Booch et al. 1999) provides building blocks of classes, interfaces and components. The available actions or methods in a class provide some level of detail, including information needed to make a service request, and information returned in response to a request. However this encodes information about access to a service rather than it’s semantics or context. Quality of Service (QoS) must be encoded at a level where actors or entities are represented, and QoS constraints can be attached to links between them. Consider then the vocabulary of these QoS constraints. Since providing or accessing a service requires technological capabilities or resources, these can provide a basic vocabulary for such constraints. One way of dividing these resources is to separate them into *computation*, *communication* and *data storage*. Taking this approach, we differentiate between those resources provided directly by an artefact and those which are provided by a lower level service, and are simply accessed by the artefact. If we have available a model of content, this enriches the vocabulary of QoS constraints. We might also wish to encode constraints between a content model and the "real−world" model. For example, that the content model reflects state of world to within some time interval. Soft real−time constraints are common in interactive systems, and particularly so in ubiquitous computing. Time constraints are relatively easy to encode (if not to meet). In the UML for example, we can annotate a sequence diagram with timing marks, and use these timing marks in constraints (see figure).

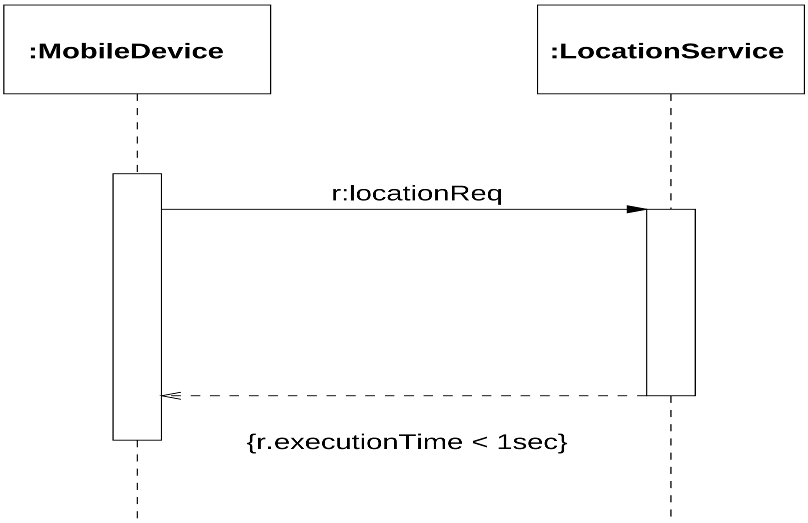

Figure 2 − Sequence diagram with timing mark and constraint

Identifying some QoS issues requires that we know details of the physical and informational distribution of the application. For example, in a UML deployment diagram where the nodes represent physically distinct entities, the links constitute communication channels with which there are associated QoS constraints. These links may themselves be stereotyped; for example as <<10BaseT ethernet>>. Where satisfaction of quality of service requirements cannot be guaranteed, the designer should consider whether support is required in the interface for dealing with degraded or intermittent service.

Discussion A user−level service definition can be seen as an encoding at a concrete level *what* an application does. This form of model is also concerned with *how* functionality can be provided by exploiting a number of services, and issues like mobility can be seen as an aspect or dimension of service provision. A service based model allows us to consider services provided to user and to appliances independently of the underlying implementation. This helps us achieve good separation of concerns, which can be very useful when moving towards concrete implementation, particularly where some ubiquitous information or communication utility is to be accessed. When constructing new applications, service based modelling may also help to increase use of existing services rather than duplicating functionality for each application. The notion of *quality of service* is of paramount importance in the area of networking. We expect quality of service analysis to be applied to an increasingly broad range of issues, including information, in which case a service based viewpoint is useful in considering both the requirements and performance of a system, since it allows the designer to consider issues concerning both the content and provision of a service. From the point of view of human factors reasoning, a service based model would be most useful when we already have detailed requirements from analysis of user tasks or activities. Meeting these service requirements then becomes the focus of analysis.

2.3 Information Based Models

Another approach to modelling is to look at the information which is stored and manipulated by the system. Such models can provide a useful basis for human factors analysis. For example the cognitive science approach of distributed cognition (Hutchins, 1995) focusses on the information used and communicated in performing a task or activity, and particularly the *representation* of this information. Since ubiquitous computing applications are likely to use modalities and representations of information which differ from those in a traditional interface, it is worth modelling the information stored, transformed and communicated, and how this affects how the information is applied in the user's task This form of modelling is concerned with what the application is to provide, by focussing on characteristics of the information or content.

Quality of information (QoI) concerns are of utmost importance in applications where diverse and distributed information sources are exploited. This can be on a small scale, as in opportunistic exploitation of locally available data (eg. from environmental sensors), up to the largest scale, as in the massive science databases for physics, materials science, astronomy and so on, which are behind recent "computational grid" initiatives. (Prime and Wilson, 1996) list a number of concerns related to QoI, such as availability, quantity, recency, measurement accuracy (eg. for scientific data), response time, and cost. Consider the QoI criterion of recency for example. This can control both the usefulness of the data for a particular purpose, access controls to data (for example, in a large scientific facility access to experimental data from high energy physics experiments might be restricted for two years in order to give those conducting the experiment time analyse the data and publish results), and cost (for example, information relating to the stock market may have a price determined by it's recency).

The ability of an application to adapt and cope with different contexts (see Abowd and Mynatt, 1999, for a discussion) depends entirely on the available information. With appropriate stereotypes, we can model syntheses of new information streams by combining others (sensor fusion). Higher level transformations, such as from sensed information to a model of user activity would require care and more complex stereotypes since they touch on the issue of user modelling. Abowd and Mynatt also raise the issue of *representations* of context, and the need to separate the sensing of context and programmable reaction to that context.

2.3.1 Static and Dynamic Information Resources

Sensed information is a central component of many ubiquitous computing applications since it is the basis for context awareness. From a modelling perspective we can distinguish sensed information as dynamic information. Consider as an example an application in which many objects in an environment have some information storage or processing capabilities, for example an application based on active badge technology (Want, 1992). Objects have attached a badge to facilitate information processing involving the object. One can imagine two types of information which can be associated with such a badge, with an associated level of functionality which may be provided on the basis of that information.

- Firstly, we have the lowest level where each badge has an associated identity. This means the system can know what the object is. We could have this concept include all *static information* about the object.
- Alternatively, we can have some capability for sensing state. This means the system can

know something about the *current state* of the object. This information can include generic attributes such as location, or type specific attributes such as sell−by dates. That is, we have *dynamic information* about the object.

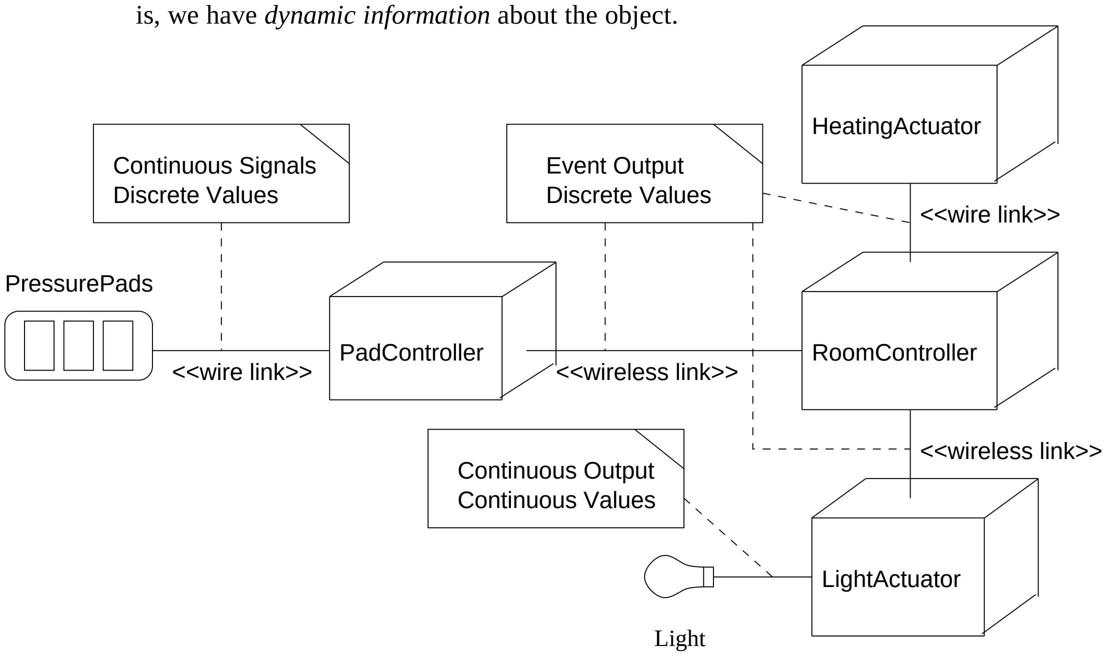

Figure 3 − Deployment diagram with stereotyped links and annotations

The frequency with which dynamic information is updated is a quality of information parameter. For instance, in the "active bat" system described in (Harter et al., 1999) the *Location Quality of Service* associated with an object (tagged with ultrasonic location transmitters) is the requested interval between location updates for the object. For dynamic information we may also have continuous or discrete update of information (eg. from a sensor). In the UML, we might encode discrete updates by means of events, and continuous updates by means of links (see Doherty et al. 2001). Connections in deployment diagrams may represent either, unless stereotyped. Additionally, stereotypes for sensors and actuators may have associated constraints on connections (eg. the nature of an input from a sensor or a control signal to an actuator).

2.3.2 Information Taxonomies

(Baber et al., 1999) propose a scheme for ubiquitous computing applications by which information is characterised along two axes; firstly a time dimension of stored, current and predicted information, and secondly a number of "reference" or conceptual dimensions. Some of these reference dimensions are design concepts such as task and application, others can be seen as high−level implementation models such as those concerning artefacts and events.

- events; activities or situations that occur in a defined time period
- task; specific activities that a person is performing (eg. instructions or procedures)
- environment (spatial and physical characteristics)
- person (self and others, eg. physiological information)
- artefact (an object in the world which the computer can recognise)

Such a framework allows us to relate the information manipulated by the application to both the user’s world, and the user’s activities within that world. This scheme does not in itself constitute what we would think of as a design model, but coupled with a model of information flow through a system (or collection of artefacts and one or more users), it allows us to take an information centric view of the users activity (whether we view this in terms of cognition, problem solving or task) when interacting with an application distributed across a collection of artefacts.

Consider as a simple example, a scenario where we have an intelligent environment, which adapts conditions (lighting, heating for example), based on the preferences of a user. Illustrated in the figure above is a collaboration diagram (with sequencing information) which models the response to a given user action in such an environment. There is a surprising amount of information either explicitly or implicitly encoded in the above. Let us briefly go over the information in this model with respect to the above taxonomy. Events: *user enters room*. In this case, since there is no recording of user behaviour over time, and no anticipation of future user behaviour, this is *current* information only. Task: User tasks are not included in the model; the task facing the system is to adapt the environment to match the user preference. Environment: the user’s presence in the space; in addition to current information, there is implicitly past information, since in this simple model there is no resetting of preferences when the user leaves the room. Person: communication with the user agent yields current person−specific preferences. Artefact: in this case we have two, since the recognition that the user has entered the room, and communication with the user agent are independent. At the information level, it is not clear whether a distinction should be made between a physically embodied user agent (eg. in a badge) or a purely digital one.

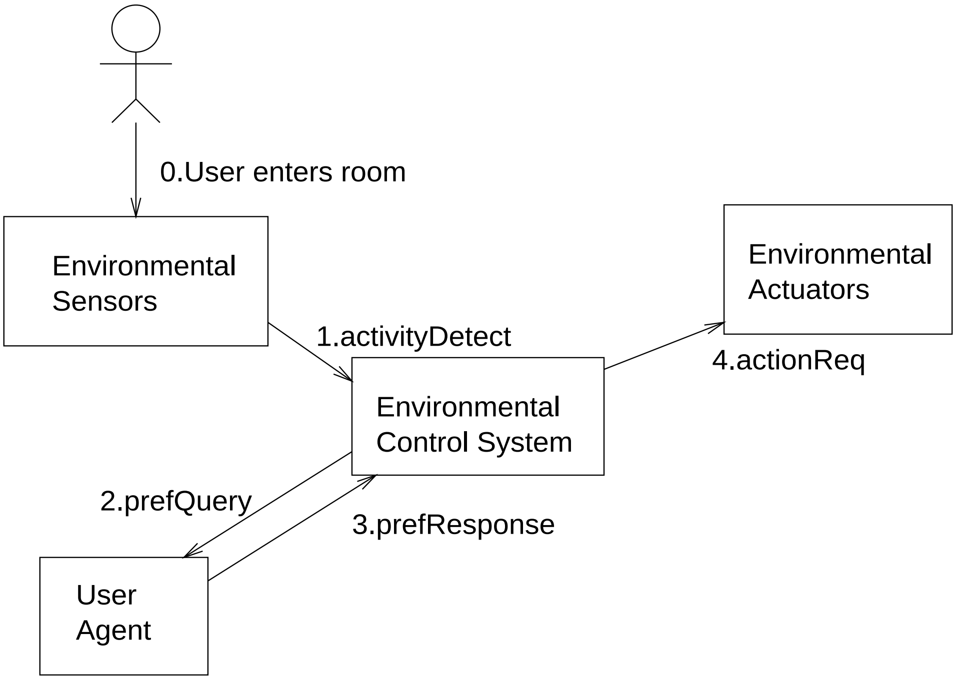

Figure 4 − Collaboration diagram illustrating interactions with sequencing information

Such a taxonomy provides a useful basis for taking a structured approach to reasoning about the information involved in a given system design. Of course it can be applied to other aspects of a design than that illustrated above, and could be used to relate different views and models of a design.

Modelling Information Flow and Quality of Information

In (Ulmer and Ishii, 2000), the issue of modelling is discussed, and a behavioural approach, at the level of MVC is described. We do not consider behavioural models for the moment, although the approach does capture information flow. However, much of the discussion concerns the coupling of artefacts and information, which is an important issue for many ubiquitous computing applications where the physical form factor plays a strong role in interaction. A number of broad system types are identified, differing in the nature of this coupling, and a large number of applications are considered with respect to these types. The distinctions between the types (spatial, constructive, relational and associative) can be made manifest in models combining control abstractions and physical sensors. However the main value of the work with respect to our immediate concerns is in providing a number of high level design options. An information model, such as one based on the taxonomy of (Baber et al. 1999) can potentially form a link between a task−based model (as might be encoded in a set of use−cases) and such detailed structural and behavioural models.

Above, we have briefly looked at the concept of quality of information, and at the issue of sensed information. We have also looked at a scheme for characterising the information manipulated by an application which can help structure information based models. Ultimately such models should help the designer to decide (and record decisions about) what information the user needs in order to carry out the desired tasks and activities, and what quality of information is required.

2.4 Physical Distribution of Information Storage and Processing Capabilities

The third approach we examine are those models which focus on the *physical distribution* of both information and processing capability. That is, is the information or capability located only within the object itself, or is it also stored within some separate information system. This requires commitment to many details of *how* a system is constructed, in order to meet requirements on *where* and in what manner an application can be used. Information regarding the physical distribution of an application is often present at a high level of a design. For example, in a use case assumptions may be made about the mobility or context sensitivity of an application. In the local or co−located case, an object might contain an encoding of both it's type and identity. This might even be built into the object (for example we can see the product bar codes in use today as an encoding of object type). Alternatively, the object might store only a unique numeric identifier, while a remote database contains a mapping between such numeric identifiers and information about the object.

In the taxonomy of (Underkoffler, 1997) *spatial context* concerns the degree of association between the information provided by the application and the physical location of the user. The manner in which spatial context which is included in an application is highly dependent on the implementation architecture. In (Harter et al., 1999) for example, location is considered in terms of geometric containment, and the link between this and the information presented is encoded as a set of prepositional rules for simple context−sensitivity. However, given current advances in the technology, it is expected that a more fluid, dynamic approach based on service discovery is likely to be commonplace, with the disadvantage however that it may be difficult to predict how the application will behave in different contexts.

In a comprehensive taxonomy covering distinctions of location and space, (Dix et al., 2000) raise the issue of physical and virtual location. This distinction is clearly illustrated by applications such as the marbles answering machine (Crampton−Smith, 1995), and many subsequent systems, in which information and computational semantics are attached to physical objects with no electronic components. Another of the distinctions discussed in Dix et al. is that of topological vs. metric (eg. Cartesian spaces). Although tied to the technology, the significance of this distinction goes beyond the granularity of location awareness since the underlying concepts (which the user may need to have an appropriate model of) are very different. As with the information taxonomy of (Baber et al, 1999), the taxonomy of Dix et al. does not in itself provide a means for encoding designs in a model. They do however provide a general object model for different kinds of entity spaces, which is used as the basis for a concrete implementation for supporting spatial context. Physical distribution tends to be reflected in structural models and part of the modelling problem in a ubiquitous computing context is the increasing importance of environment. Coupled with this, mutable physical attributes can be an important part of the state and external representation of the system. Some special purpose models have been developed to address mixing physical location issues with other design issues. An example is the Environmental Interactive Systems approach of (Watts et al., 2000), which examines the coupling of distributed physical and digital entities, with a focus on collaborative work and shared representations.

Direct support for reasoning about this kind of issue is sparse (in the UML for example, it is limited to *location* attributes, and the use of stereotypes in deployment diagrams which imply independent artefacts, as in the diagram above). Part of the difficulty is the continuous and mathematically involved nature of the underlying technology, such as wireless signal transmission by infrared beam, or by microwave signals from mobile phones.

2.4.1 Mobility

Once we have modelled the distribution of information and processing capability (and hence the services provided via this capability), we can look at the issue of mobility. Mobility seems to be one of the most compelling factors influencing the adoption of ubiquitous computing appliances. One can look at mobility in terms of continuity of service. Two interesting design issues with respect to mobility are:

- *location of service provision* When the service is accessed via the mobile device;
    - it is possible that the service is provided entirely by the device,
    - that the device simply provides an interface to some utility
    - or that the service may be provided by a distributed group of artefacts.
- *context awareness and adaptation*
    - location − Is the device "aware" of it's location, and hence can it provide services tailored to it's current geographical context, see eg. (Spohrer, 1999, Abowd and Mynatt, 2000).

− service discovery − Is the device aware of other devices and services within its current context; ie. is the device capable of some form of service discovery.

A number of mobility issues concern the location of service provision. Firstly there is a range issue; if the application relies on a utility or other artefacts, then it depends on a communication link (of a certain quality, for example in terms of bandwidth or latency) being available. Likewise, failures in the other artefacts or the utility can have implications for the services provided, and complicate the possible failure modes of the application. One view on this issue is to see it as a continuum between *user−borne* and *environmental* systems (Underkoffler, 1997). Wearable computers and smart environments can be seen as delineating the extremes of this axis, with other approaches lying somewhere in between.

In terms of location awareness, we must consider both the quality and granularity of location data, and the quality of associated information. For example, do we have metadata associated with geocoded information, allowing context sensitivity beyond location?

2.4.2 Position and Proximity

It is expected that much of the interface of some ubiquitous computing applications will be "tangible" or rooted in the physical characteristics of objects and environment. Thus whether we focus on the sensors+devices view of "sentient computing" (Hopper, 1999) or the "tangible interfaces"(Ishii, 1997) view of physical manipulation and feedback, much of the interface will involve the position and orientation of objects (including parts of the users own body) as well as proximity to other objects and devices. Yet another open question is how to go about designing interactions based on these physical attributes. (Hopper, 1999) refers to this as the challenge of "programming with space".

Proximity can have a significance for an application beyond triggering of application actions. In the active bat system there is an "action zone" around a workstation, but beyond this there is a "maintenance zone", for which the application maintains an awareness of contained objects. In the "teleport" application, the workstation has ready the workspace of users in the maintenance zone, such that the appropriate workspace can be displayed if a user enters the action zone of the workstation.

Modelling to support examination of these issues is difficult, not least because the physical interpretation of positioning and proximity seems very often to be determined by the constraints of the technology. Part of the problem is that technological constraints still have a strong influence. For example, (Harter et al., 1999) map to a geometric containment model from a 2−D view using elliptical regions yielded by the underlying technology. This may work well for the ultrasonic "active bat" location system, but it is unlikely to be appropriate for systems based on other technologies such as radio−frequency identity chips, or for very directional systems such as infra−red beams.

2.4.3 Information Flow

When we have modelled the distribution of information and computation, or of service provision, we can look at how information flows through the system, and particularly at the interactions of the user both "directly" with information appliances or artefacts and "indirectly" with the environment or physical processes within the environment.

Consider for example an interactive whiteboard − the MagicBoard application (Berard, 1999) (see figure below). The user draws directly on an ordinary whiteboard; what is drawn on the board is scanned by the system. Additionally, the user can use gestures to perform operations on what is drawn on the board, the results of which are projected on the board as "digital ink", additionally a tracking cursor indicates where the system thinks the user is pointing. Thus, there is gestural input from user to system, and from system to environment by means of the projector, and direct observation of the environment (the board) by the user. However we see that there is also direct input from user to environment (by means of drawing on the board), and also from environment to system (from scanning images on the board).

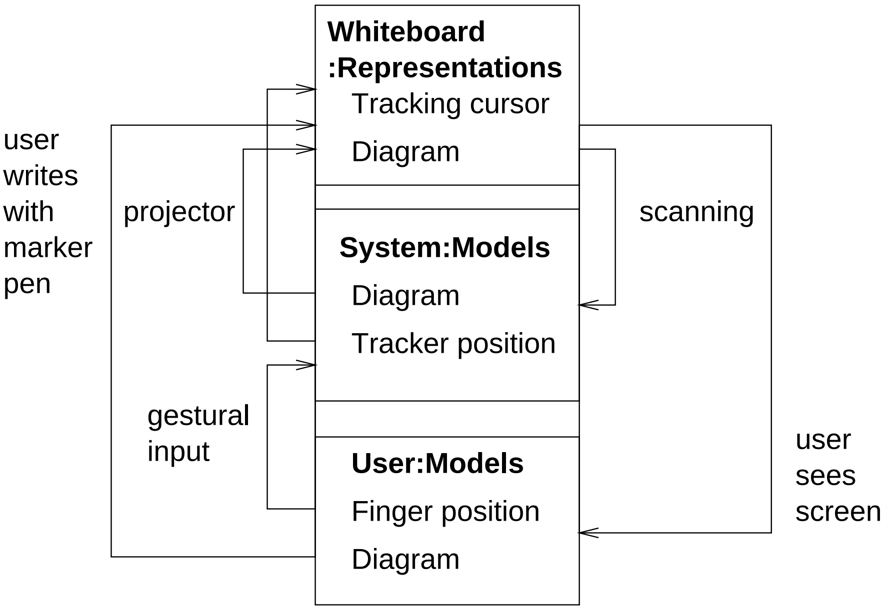

Figure 5 − Information flow

By modelling these links (which ultimately must have some realisation at the physical level), we can look at issues of environmental and control noise, at failure modes when particular links or channels degrade or become unavailable, and at the different quality of service parameters (such as latency) which apply to these links and which might have an impact on usability. This issue of the flow of information is also of importance for consideration of security and privacy, particularly where personal or financial information is communicated.

2.4.4 Distribution of Information Sensing and Processing

For a model including physical distribution aspects to be useful it must necessarily incorporate other design information. Given the particular importance of information sensing, let us consider the sensing and distribution of information through a ubiquitous computing application, such as an active badge system. Useful interactions within the system would likely require at least some static information, particularly type and identity information. At the next higher level, we have the sensing of state information. This information can be sensed and communicated explicitly by the objects themselves (eg. whether the badge is being exposed to light) or may need to be sensed by some external entity (as might be the case with location for a badge which emits an identifying signal periodically). Likewise with "active" objects, the processing can be located within the object, or implemented by some remote computing service; for example, if the "code" of the agent were stored in some device (eg. a smart card containing java code) but operation of the agent requires some host device with computing capability.

			Co−located	**Remote**
Information	**Static**	**identity** **type** **static attrs**	***RF ID badge*** ***Barcode***	***Recognition systems***
	Dynamic	**position** **temperature** **ownership** **object−specific**	***GPS receiver*** ***Smart card***	***Camera tracking***
Processing	**Passive**	**continuous activity**	***biosensing devices***	***Smart environments***
	Active	**(different processing capabilities)**	***Wearable computers***	***Environmental agents***

The table above plots distinctions between information and processing capabilities against their physical distribution, with examples of technologies for each case. (Underkoffler, 1997) calls the distinction between active and passive processing the *input* dimension. This concerns whether active attendance is required or whether a passive user still has significant interaction with the application (as might be the case with a location aware system). From a modelling perspective, these different categories are candidates for stereotypes.

Modelling Physical and Digital Distribution

The question which construction of such a model should help to answer is whether the physical infrastructure is in place to deliver information at the right place and time. An important property with respect to the physical distribution of information concern the time taken to deliver information after the user has developed a need for the information. In some cases, the user must articulate this need; in other cases it might be sensed by the system. Transmission between physically distinct locations may take an appreciable amount of time, particularly if connections must be initiated, or where the infrastructure or environment are not conducive. In some cases this may require some anticipation of the users information requirements. Another serious issue concerns failure or degradation of elements of a ubiquitous computing application, or of links between them. While a ubiquitous computing application may give us the advantage of independent failure modes, allowing a system to continue to work (although perhaps at reduced capacity) following the failure of some component, the number of components and links potentially involved, coupled with changing environments and infrastructure make such failure all the more likely. Where reliability is of any importance, rigorous analysis based on a physical distribution model is a necessity.

3. OTHER ISSUES

In the above, we have looked at three general approaches to modelling, and support for these approaches. Here we briefly mention some other issues particularly relevant to future computing systems.

3.1 Lifecycle Modelling

Ubiquitous computing applications based around a number of independent artefacts face another challenge since it is likely that they will be added, maintained and replaced at different times. Both physical and informational artefacts have an associated lifecycle. In each case the designer should consider the overheads both in terms of user effort and cost. Again we see issues of continuity, such as that of an application, independent of the components and services which implement it.

To complicate the situation further, entities may be involved in a number of different applications, which may reduce lifecycle costs.

− instantiation or initialisation, associating entity identifiers with information, entry of information, configuration, and so on.

− maintenance, ensuring consistency, updating information, reconfiguration to accommodate new applications, recovery from failures.

− expiry, removing information associated entirely, or declaring it defunct in a way which does not lead to consistency problems.

To use the most common example, consider a simple physical artefact in a coffee dispensing application − a coffee mug with an associated owner. Where an artefact (such as a mug) may be associated with different people over their lifetime, there are issues of privacy, particularly in the case of a distributed application, where information might take time to propagate through a system. We show in figure 6 an activity diagram of a simple system, which in this case captures the lifecycle of a coffee mug. Note however that the more entities which interact within the system, the more difficult it becomes to simultaneously capture the lifecycle of different objects. In this case, there is no straightforward way to reflect the lifecycle of objects in the database in the same diagram, as we are potentially dealing with multiple consecutive owners of the same mug, multiple mugs owned by the same owner, changing owner account details, and so on.

To support more in−depth reasoning about this issue, we would have to add an explicit time dimension (such as expected lifespan), consideration of transient elements of the application architecture, replaceable infrastructure, and high−level issues such as the various strategies by which tasks may be carried out, and organisational issues such as standard procedures, responsibility and job design.

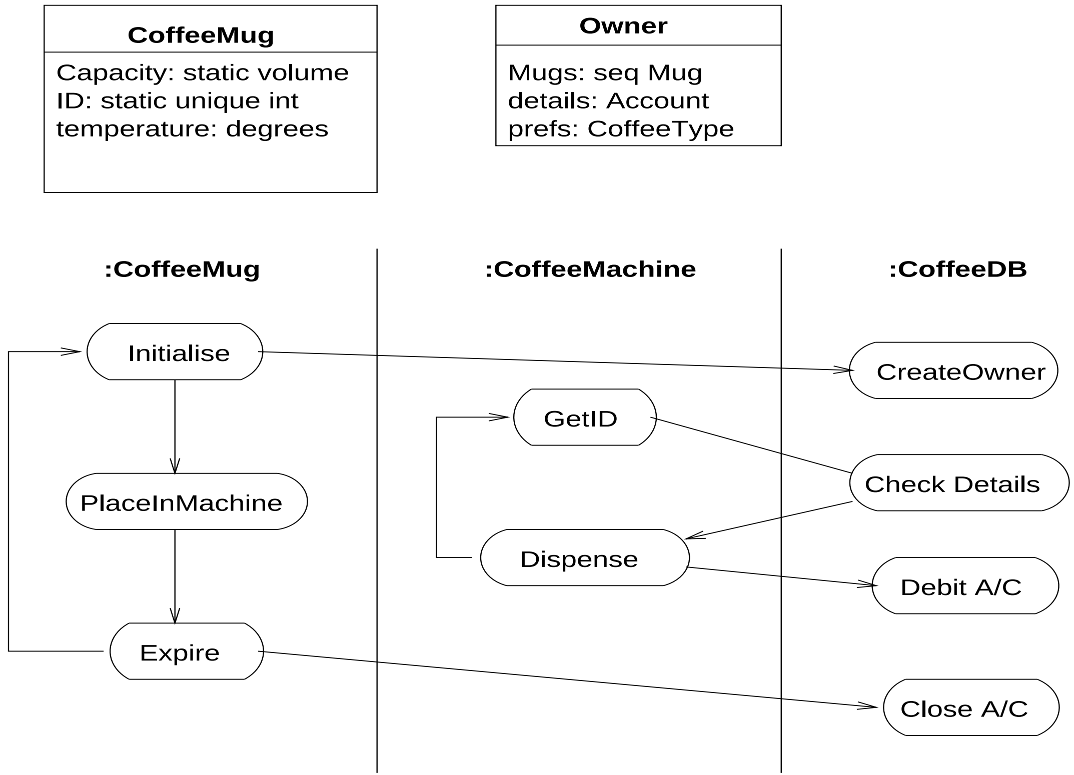

Figure 6 − Activity diagram of artefact lifecycle

3.2 Behavioural Models

If we wish to look at detailed, dynamic situations and sequences of behaviour, then behavioural models can be used to support our reasoning. Simple formalisms can be used to capture sequencing and possible modes of use of the system by the user, and more complex formalisms can be used to capture real−time behaviours and complex functionality. Two modelling approaches which may be particularly applicable in this area are hybrid and stochastic models. Hybrid models allow us to describe complex, continuous system and environmental processes, in particular looking at real−time properties of interactions (Doherty at al. 1999). Stochastic modelling techniques allow us to include statistical assumptions about the performance of both system and user in the model, generating performance results for operation of the system by the user which are themselves statistical in nature, giving us performance estimates which are a much richer resource for reasoning (work in progress).

3.3 Human Performance

As has been stated above, a potential advantage of ubiquitous computing applications is that they can make use of a wide range of sensory modalities for communication with the user. However, simply using multiple modalities does not automatically deliver performance improvements. In order to effectively design multimodal interfaces, one must consider the user’s ability to process and integrate input along different sensory channels, and also to produce output via different motor and articulatory channels. In (Schomaker, 1995) a substantial investigation into the human information processing involved in multimodal interactions is presented. There, observations by the user are considered at a number of levels:

- physical or physiological level
- information theoretical
- cognitive level
- intentional level

Some physical aspects of interactions have been discussed in the models above, but a common framework for a variety of physical stimuli is absent. Low level physiological interactions between the senses are considered in (Welch, 1986), and some sensitivity data is also presented. With regards to the models discussed above, some coverage is achieved at a cognitive level, looking at representational and procedural aspects. The intentional level (dealing with goals and beliefs) is the topic of much traditional work in HCI. One must also consider assumptions about user abilities embedded within a given interface design. This does not just concern those with sensory and motor impairments, but diverse environments and changing settings of a ubiquitous computing application may not always be amenable to interaction via a given modality.

3.4 Higher Level Issues

At a higher level we have collaborative issues such as *participation* (Underkoffler, 1997), whether a system prohibits, permits or encourages collaboration between users. There is something of a gap in the models discussed above when considering collaborative use. This is a challenging issue in ubiquitous computing given that users may be co−located or remote (or a combination of these), fixed or mobile, collaboration might be at most between two users, or might allow many, and so on. A service definition might include details of communication with remote systems, as well as the number of local clients of that service, but this level of abstraction misses out the social context of collaboration, shared knowledge and representations, and so on. The Environmental Interactive Systems approach of (Watts et al., 2000) goes some way towards addressing these issues.

3.5 Technology Acceptance

Combining a lifecycle model with an object based model in which information and processing capability is assigned to given artefacts may allow us to look at the issue of technology acceptance. As the enabling technology for each artefact matures, the overheads associated with the initial cost of the artefact decrease, but the lifecycle overheads are likely to remain unless explicit measures are taken to alter them.

4. CONCLUSIONS

It is clear that with the widespread deployment of ubiquitous computing devices, incorporating information sensing, processing, storage and communication capabilities, the design space of interactive systems will be massively expanded. However, along with the greater freedom available to the designer, a far broader set of issues need to be considered, including physical properties of objects, communication and co−operation between "information appliances", exploitation of communication and information infrastructures, mobility and contextual information. Correspondingly, we must take into account the capabilities and needs of users in these new settings in support of a wider range of tasks and activities.

Work on general user interface concerns, for example related to task modelling, are also relevant, but here we have tried to focus on those aspects of ubiquitous computing applications which are novel and different. We have considered three general approaches to modelling ubiquitous or pervasive computing systems, and some of the modelling issues which arise with them. Service based models have a strong architectural component, and are important when considering whether basic quality of service requirements can be met, particularly when the system comprises a distributed and heterogeneous set of components. Information is the lifeblood of an interactive system, so it is important to consider what information is needed by the user to support tasks and activities in a given context. In a ubiquitous computing setting dynamically sensed information is particularly important as it is the basis for context sensitivity, which itself allows continuity with respect to task and activity.

(Abowd and Mynatt, 2000), give a challenging research agenda for ubiquitous computing. Restricting ourselves to the topic of modelling alone, we would put forward the following items:

- Conventions and stereotypes for modelling services, tasks (a general interactive systems modelling concern), and information (perhaps an instantiation of the taxonomy of Baber et. al.).
- Models (and stereotypes) of standard components for sensing, and for sensor fusion.
- A framework and vocabulary for expressing quality of information constraints, and particularly for constraints with an explicit time dimension.
- Conventions for modelling location and mobility issues. A *location* attribute is hopelessly inadequate for capturing such details. The taxonomy of Dix gives a starting point; an important dimension of this is separating the spatial model of the underlying technology, that which underlies the application from the user perspective, and possible transformations between the two.

To summarise, we have examined some of these issues, and models which can help the designer in considering these issues. We have looked at encoding models in a standard notation to help answer specific design questions of particular relevance in this application area, and listed a number of modelling issues which should be resolved.

REFERENCES

C. Baber, D. Haniff and S. Wooley. "Contrasting paradigms for the development of wearable computers", IBM Systems Journal, 38(4), 1999.

F. Bérard. "Vision par ordinateur pour l’interaction fortement couplée", PhD thesis, L’Université Joseph Fourier, Grenoble 1, 1999.

G. Booch, I. Jacobson and J. Rumbaugh, "The unified modeling language user guide", Reading, Mass. : Addison−Wesley, 1999.

G. Crampton Smith. The Hand That Rocks the Cradle. I.D., May/June 1995, pp. 60−65.

A. Dix, T. Rodden, N. Davies, J. Trevor, A. Friday and K. Palfreyman. "Exploiting Space and Location as a Design Framework for Interactive Mobile Systems", ACM Transactions on Computer−Human Interaction, 7(3), September 2000.

G. Doherty and M. Massink, "Continuous Interaction and Human Control", Proceedings of XVIII European Annual Conference on Human Decision Making and Manual Control. Group−D Publications, Loughborough, J. Alty Ed., 1999.

G. Doherty, M. Massink and G. Faconti. "Using Hybrid Automata to Support Human Factors Analysis in a Critical System", Proceedings of ERCIM Workshop on Formal Methods in Industrial Critical Systems, Trento, July 1999 (also Journal of Formal Methods in System Design, in press).

G. Doherty, T.Anderson, M.Wilson and G. Faconti. "A Control−Centred Approach to Designing Interaction with Novel Devices", to be presented at the Conference on Universal Access in Human Computer Interaction, New Orleans, 2001.

N. Gershenfeld. "When things start to think", Hodder & Staughton, 1999.

A. Harter, A. Hopper, P. Steggles, A. Ward and P. Webster. "The Anatomy of a Context−Aware Application", AT&T Laboratories, Cambridge, Technical Report 1999−7, 1999.

A. Hopper. "The Royal Society Clifford Paterson Lecture − Sentient Computing", AT&T Laboratories Cambridge, Technical Report 1999.12, 1999.

E. Hutchins. "Cognition in the Wild", MIT Press 1995.

H. Ishii and B. Ullmer. "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms", in Proceedings of CHI ’97, Atlanta GA, March 1997, ACM Press, pp 234−241, 1997.

D. A. Norman. "The Invisible Computer", MIT Press, 1998.

L. Mittag. "Innovations in Internet Appliance Design", Embedded Systems Programming 12(5), May 1999.

P. Nixon, S. Dobson and G. Lacey, Managing Smart Environments, Proceedings of Workshop on Software Engineering for Pervasive and Wearable Computing, ICSE 2000.

M. Prime and M. Wilson. "Quality of Service for Information Access". Proceedings of 2nd ERCIM Workshop on User Interfaces for All, Prague, November 1996.

L. Schomaker et al., "A Taxonomy of Multimodal Interaction in the Human Information Processing System", Report of ESPRIT Project 8579 (MIAMI), WP−1, February 1995.

J. C. Spohrer, "Information in Places", IBM Systems Journal 38(4), 1999.

B. Ullmer and H. Ishii, "Emerging Frameworks for tangible user interfaces", IBM Systems Journal, Vol. 39(3&4), 2000.

J. Underkoffler. "Antisedentary Beigeless Computing", Personal Technologies 1(1), pp.28−40, 1997.

R. Want, A. Hopper V. Falcao and J. Gibbons. "The active badge location system", ACM Transactions on Information Systems 10(1), January 1992.

L. Watts, J. Coutaz, D. Thevenin, E. Dubois, M. Massink, G. Doherty. "Environmental Interactive Systems: principles of a digital−physical fusion", TACIT Report TR−013, http://kazan.cnuce.cnr.it/TACIT/TACITweb/DOCUMENTS/TechnicalReports/TACIT−TR013.ps, 2000.

M. Weiser. "Some Computer Science Issues in Ubiquitous Computing", CACM, July 1993.

R. B. Welch and D. H. Warren. "Intersensory Interactions", Handbook of perception and human performance", Chapter 25, K.R. Boff, L. Kaufman and J. P. Thomas, Editors, New York: Wiley, 1986.